

PHP

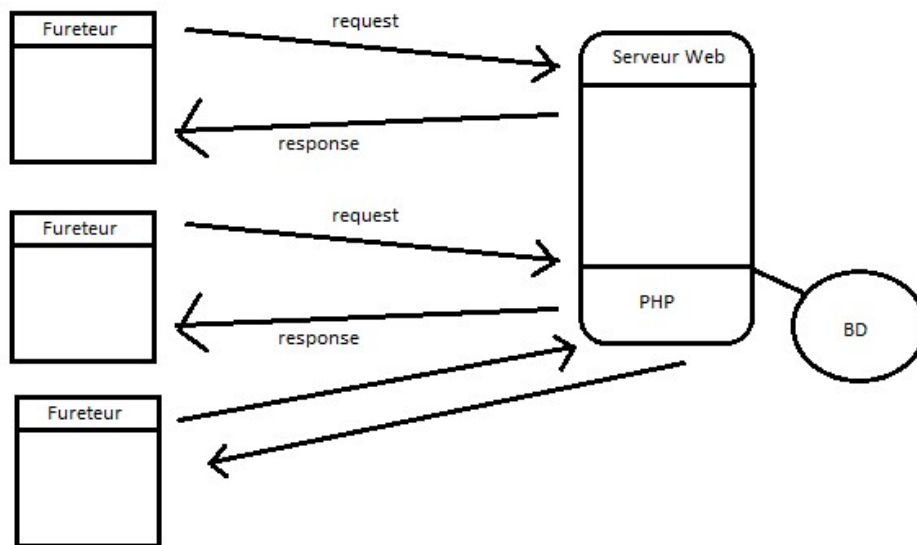
Origine

Rasmus Lerdorf avait un site web (1994) où il affichait son CV. Il était curieux de voir qui le consultait. Il a créé un petit outil à cet effet qui est devenu PHP : Personal Home Page. Le nom a évolué pour devenir : PHP Hypertext Processor.

Client/Serveur

Le Web fonctionne selon une architecture client/serveur. Les clients sont les fureteurs (Chrome, Firefox, Edge, Safari, etc) qui peuvent communiquer avec des serveurs Web. Les serveurs sont des logiciels s'exécutant sur des sites accessibles par des URI débutant par http:// ou https://

Ainsi un client fait une demande (request) à un serveur et celui-ci répond (response) au client :



Un des serveurs web les plus répandus est **Apache**. Il s'exécute d'ailleurs sur notre serveur web de production **techinfo-cstj.qc.ca**. Apache intègre un interpréteur PHP. PHP est un langage fortement associé à la programmation du côté serveur.

PHP et Serveur Web

Lorsqu'on tape une **URI sans nom de fichier** dans un fureteur Web, par exemple :

<https://amartel.techinfo-cstj.ca/4203D4>,

le serveur qui répond à ce nom de domaine regardera

- 1- Y a-t-il un fichier nommé index.php dans le dossier RacineDuDomaine/4203D4/ ?
- 2- Si oui, le fichier index.php sera traité par l'interpréteur PHP du serveur puis le résultat de cette interprétation sera envoyé au client (fureteur)

- 3- Sinon, le serveur cherchera le fichier index.html. Si celui-ci n'y est pas le serveur retournera une erreur

Lorsqu'on tape une **URI avec un nom de fichier**, par exemple :

<https://amartel.techinfo-cstj.ca/4203D4/info.php>

le serveur web du domaine regardera

- 1- Y a-t-il un fichier nommé info.php dans le dossier RacineDuDomaine/4203D4/ ?
- 2- Si oui, le fichier info.php sera traité par l'interpréteur PHP du serveur puis le résultat de cette interprétation sera envoyé à l'utilisateur
- 3- Sinon, le serveur retournera une erreur

Le PHP s'exécute sur le serveur, il n'a donc aucun accès à ce que fait l'utilisateur avec sa souris ou son clavier.

Les seules interactions de PHP sont avec le serveur Web.

Le serveur Web lui-même, ne réagit qu'à la soumission d'une URI par un client Web.

Il y a quatre façons pour un client Web de soumettre une requête à un serveur Web :

- Taper l'URI dans la zone d'adressage d'un fureteur,
- Cliquer un hyperlien : `<a href=... >`
- Soumettre un formulaire : `<form> <input type='submit'>...`
- Faire une requête asynchrone (AJAX)

Si l'URI demandée correspond à un fichier .php, le serveur Web passera la demande à l'interpréteur PHP. Le but du PHP sera de générer du HTML correspondant à ce que le client demande, la réponse du serveur sera le HTML ainsi produit.

Structure d'un programme PHP

Un fichier PHP est un fichier dont l'extension est .php. Il peut contenir du HTML pur, du PHP pur ou un mélange des deux. Le but est d'envoyer un rendu HTML au client, rendu qui sera adapté au choix et contexte de l'utilisateur.

Les blocs de code PHP dans un script débutent avec `<?php` et se termine par `?>`

Éléments et syntaxe du langage

Commentaires

D'une ligne : `//` ou `#`

D'un groupe de lignes : `/* ... */`

Variables

Les variables se déclarent facilement en PHP. Il suffit de préfixer une chaîne de caractères par un `$`.

Exemples :

```
$var1 = 0;
```

```
$var2= « Salut bonjour »;
```

Par défaut les variables n'ont pas de type défini. Une même variable peut être tour à tour un nombre entier, un réel, une string, un tableau ou un objet. Bien sûr ça inclut des tableaux d'objets, des objets contenant des tableaux, etc.

Le type d'une variable peut être donné par :

- Le contexte d'utilisation de la variable
- La dernière initialisation de la variable
- Force

Le nom des variables est « case sensitive »

Les strings

Longueur : `strlen($var2)`

Concaténation : `.`

Guillemet vs apostrophe == interprétation vs direct

Limites de l'interprétation avec guillemets

Constante

```
define('NOM_DE_CONSTANTE', saValeur);
```

Instructions

Une ligne de code en php doit **obligatoirement** se terminer par **;**

Fonctions

Les fonctions sont déclarées par le mot clé **function** :

```
function nomFonction(){ . . . }
```

Les fonctions peuvent prendre des paramètres, les paramètres peuvent être typé ou non, c'est facultatif.

Les fonctions peuvent retourner ou non et c'est facultatif de spécifier un type de retour.

Le nom des fonctions N'est PAS « case sensitive »

Portée des variables, zone globale et zone fonction

Opérateurs

Mêmes opérateurs que tous les langages descendants du C : `+` `-` `*` `==` `!=` ...

On ajoute le `===` . Pour qu'un test impliquant cet opérateur soit vrai il faut que la valeur ET le type soient identiques

Structures de contrôle :

If,

elseif,

else

while

for :

```
for($i=0; $i<count($tabElement); $i++) { ... }
```

foreach :

```
foreach ($tabElement as $unElement) { ... }
```

switch

Outils de la librairie

Echo (\$msg): affiche \$msg

Die(\$msg) : arrête l'exécution du script php et affiche \$msg

Var_dump(\$msg) : affiche le type et la valeur de \$msg

Rand(min, max): retourne un nombre aléatoire entre min et max

Isset(\$variable) : retourne vrai si \$variable a été initialisée et faux sinon

Count(\$tab) : donne le nombre d'éléments dans \$tab

Header() : modification des headers HTTP

Fichiers externes

Include(« nomFichier ») : inclut le contenu du fichier nomFichier

Tableaux

Indices automatiques

Indices numériques

Indices littéraux

Initialiser un tableau :

```
$tab = array(0=>'val0', 1=>val1, 2=>val2);
```

Ou

```
$tab = [0=>'val0', 1=>val1, 2=>val2];
```

Vider un tableau : `$tab = array()` ou `$tab=[ ]`

Ajouter un élément au tableau : `$tab[] = $neoElement;`

Compter le nombre d'éléments d'un tableau : `Count($tab)`

Boucler sur les éléments d'un tableau :

for :

```
for($i=0; $i<count($tabElement); $i++) { ... }
```

foreach :

```
foreach ($tabElement as $unElement) { ... }
```

Boucler avec `$cle=>$val`

```
foreach ($tabElement as $cle=>$val) { ... }
```

PHP et les formulaires HTML

Plusieurs variables super globales sont accessibles dans tous scripts PHP. En voici deux qui sont très utilisées : `$_GET` et `$_POST`. Ces variables sont des tableaux à indices associatifs qui permettent de récupérer les valeurs fournies par un utilisateur dans un **<form>** HTML.

Si j'ai un formulaire ressemblant à ceci :

<form>

<input type='text' name='Matricule'>

<input type='submit' value='valider'>

</form>

Quand l'utilisateur cliquera sur 'valider', le script qui offrira ce formulaire sera demandé et il pourra avoir accès à la valeur du champ **Matricule** ainsi :

```
$code = $_GET['Matricule'];
```

Par défaut l'attribut **action** concerne le même script que celui offrant le formulaire. On peut spécifier explicitement l'attribut **action** :

<form action='valideur.php'>

<input type='text' name='Matricule'>

<input type='submit' value='valider'>

</form>

Quand l'utilisateur cliquera sur 'valider', le script 'valideur.php' sera demandé et il pourra avoir accès à la valeur du champ **Matricule** de la même façon :

```
$code = $_GET['Matricule'];
```

Un autre formulaire pourrait spécifier l'attribut **method** (par défaut elle est == **get**) ainsi :

```
<form method='post' action='valideur.php'>
```

```
    <input type='text' name='Matricule'>
```

```
    <input type='submit' value='valider'>
```

</form>

Quand l'utilisateur cliquera sur 'valider', le script 'valideur.php' sera demandé et il pourra avoir accès à la valeur des inputs du formulaire en passant par \$_POST ainsi :

```
$code = $_POST['Matricule'];
```

PHP OO

Les classes

Les attributs

Les méthodes

Le constructeur

Instancier un objet

Utiliser ses méthodes avec ->

L'accès aux attributs et méthodes par \$this->

API MySQLi

PHP et MySQL : un couple étroitement uni

Accès à une base de données

Requête en lecture

Traitement du résultat

Requête en écriture

Communication client/PHP

Passer des paramètres à une URI

<http://pokernirvana/partie.php?numpartie=18>

le \$_GET

Autres super globales

Configuration de PHP

Transformation d'information en JSON

Syntaxe JSON

Fonction JSON_Encode

Les entêtes des réponses HTTP

Pour indiquer au client le type de fichier retourné et autre info, par exemple concernant les autorisations

Configuration de PHP

Php_info()

Ini_get

Ini_set

Timezone

MAX_EXECUTION_TIME

La mise en production d'un site web php

Transfert ftp

PHP dans un autre contexte : CLI

Php au prompt

Framework PHP

Symfony

Architecture MVC